Errata Report

mP6™ MICROPROCESSOR FOR WINDOWS™ SYSTEMS

Doc#: MKTER6401.00
Version 0.0

Devices sold by Rise™ Technology are covered by the warranty and patent indemnification provisions appearing in it's Terms of Sale only. Rise™ Technology makes no warranty, express or implied or by description regarding information set forth herein or regarding the freedom of the described devices from patent infringement. Rise™ Technology Company reserves the right to discontinue production and change specifications and prices at any time and without notice.

Rise™ Technology assumes no responsibility for the use of any circuitry other than circuitry embodied in a Rise™ Technology product. No other circuits, patents or licensees are implied.

Rise™ Technology products are not authorized for use as components in life systems or as part of life support devices.

INTRODUCTION

This newsletter contains errata for the Rise™ *mP6*™ processor. For the purposes of this newsletter, errata are best described as differences between anticipated and actual behavior of the Rise™ *mP6*™ processor.

SUMMARY TABLE OF ERRATA

Table 1 summarizes the Rise™ *mP6*™ processor errata and provides stepping, status and future plans for each erratum. The stepping of the Rise™ *mP6*™ processor is identified by the stepping code returned by the CPUID instruction, or in the EDX register following a Reset.

Codes used in Table 1:

Found: Erratum is found in the stepping as shown

Fixed: This erratum is fixed in the stepping as shown

Fix: The erratum applies to this stepping and will be fixed in a future stepping

No Fix: There are no plans to fix this erratum

Table 1. Summary Table of Errata

No	Errata Status			Errata
	Found	Fixed	Plan	
1	4.0	4.1c		INIT ignored in HALT state in the Windows NT* Environment
2	4.0	4.1b		Bus Ratios at 3X and 3.5X Don't Work on Stepping 3.0 Parts
3	4.1b	-	No fix	STOPCLK before INIT causes hang
4	4.0		Fix	24-bit and 53-bit precision mode FMUL operations produce incorrect results when both operands consist of higher precision numbers than the precision mode

ERRATA

1. INIT Ignored in HALT State in the Windows NT* Environment

PROBLEM: The system hangs on Windows NT when attempting to execute the restart option.

IMPLICATION: If the Rise™ *mP6*™ processor is in HALT state when the INIT pin is activated, it remains that state following the INIT. In other words, the Rise™ *mP6*™ processor ignores the INIT pin when the processor is in the HALT state. This prevents the Rise™ *mP6*™ processor from taking the INIT exception and starting the INIT algorithm, which results in the INIT pin being asserted but not initializing the Rise™ *mP6*™ processor.

WORKAROUND: Push the RESET button to make it reboot the system.

STATUS: A fix has been implemented in the 4.1c production stepping of the Rise™ *mP6*™ processor.

2. Bus Ratios at 3X and 3.5X Don't Work on Stepping 3.0 Parts

PROBLEM: All stepping 3.0 parts able to run Windows* 95 @ 100Mhz x 2, fail to run @ 66Mhz x 3. Also, parts that can run @ 100Mhz x 2.5, fail @ 66Mhz x 3.5.

IMPLICATION: It turned out that some critical paths exist in the 3X and 3.5X circuits which caused the clock synchronization to fail to boot–up at 3X or 3.5X bus ratios at the expected full speed.

WORKAROUND: Use 2.0X or 2.5X ratios.

STATUS: A fix has been implemented in the 4.1b stepping of the Rise™ *mP6*™ processor.

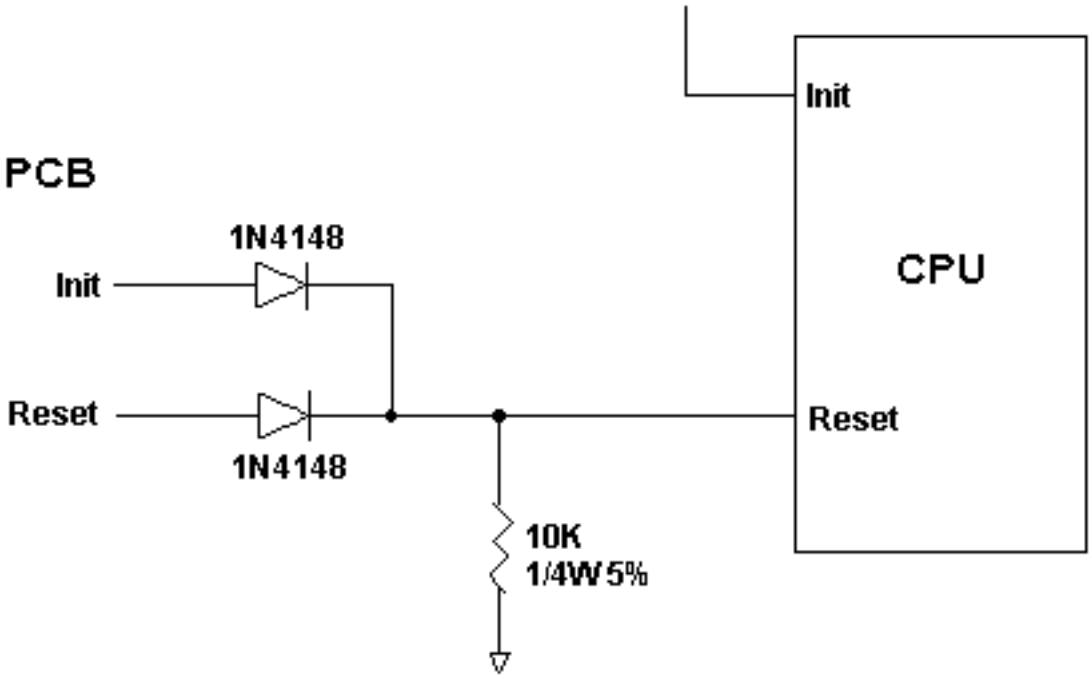
3. STOPCLK before INIT causes hang

PROBLEM: System fails to boot with Rise™ *mP6*™ processor when STOPCLK is active during or shortly after RESET drops.

IMPLICATION: This problem has been observed only on the SMSC socket 7 motherboard with SLC90E42 Northbridge and SLC90E46 Southbridge at 2X mode; it has never been seen on a Super Socket 7 chipset.

WORKAROUND: Use a 2.5X ratio, or Run BIST at power on (refer to the modification below).

STATUS: There currently are no plans to fix this erratum. Expecting SMSC to change future chipset revision such that STOPCLK is not active during RESET.



4. 24-bit and 53-bit precision mode FMUL operations produce incorrect results when both operands consist of higher precision numbers than the precision mode

PROBLEM: When an FMUL operation is executed as described in the following cases, the FPU generates an incorrect result:

Case I

1) Precision control is set to 24-bits
2) Rounding mode is set to “round up” or “round to nearest”
3) Both operands consist of more than 24 leading 1s in the significand field (i.e., operands are larger than the largest single precision number can represent)

Case II

1) Precision control is set to 53-bits
2) Rounding mode is set to “round up” or “round to nearest”
3) Both operands consist of more than 53 leading 1s in the significand field (i.e., operands are larger than the largest double precision number can represent)

In each case, the Rise™ *mP6*™ processor produces a result that has an unsupported format in the FPU internal 80-bit representation. Depending on the precision setting, the result is converted as shown here

- For 24 bit precision

  The expected result should have a significand equal to: 1.0000…..0000 (23 zeroes)

  The Rise™ *mP6*™ processor produces a significand equal to: 0.0000…..0000 (23 zeroes)

- For 53 bit precision

  The expected result should have a significand equal to: 1.0000…..0000 (52 zeroes)

  The Rise™ *mP6*™ processor produces a significand equal to: 0.0000…..0000 (52 zeroes)

IMPLICATION: Impact is negligible.

The 24-bit and 53-bit significand modes were designed to conserve the data-memory of the operands. FPU operations in any precision mode using operands of higher precision are highly unlikely. The internal operating precision of the FPU accommodates a 64-bit significand.

WORKAROUND: None

STATUS: A fix will be implemented in a future production stepping of the Rise™ *mP6*™ processor.